

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

REALTEK SEMICONDUCTOR CORP.,  
Petitioner,

v.

ATI TECHNOLOGIES ULC  
Patent Owner.

---

Case No. IPR2023-00564  
U.S. Patent No. 7,742,053

---

**PATENT OWNER'S SUR-REPLY TO PETITIONER'S REPLY**

## TABLE OF CONTENTS

|                                                                                    | <b>Page</b> |
|------------------------------------------------------------------------------------|-------------|
| I. INTRODUCTION .....                                                              | 1           |
| II. CLAIM CONSTRUCTION .....                                                       | 1           |
| A. Storage of “command threads” .....                                              | 1           |
| B. “at least one memory device . . .” .....                                        | 2           |
| C. “select a command thread . . .” .....                                           | 3           |
| D. “command processing engine” .....                                               | 3           |
| III. STUTTARD .....                                                                | 4           |
| A. No Storage Of “Command Threads” At The Same Time, In<br>Separate Portions ..... | 4           |
| 1. Phased Operation .....                                                          | 5           |
| 2. SIMD .....                                                                      | 9           |
| 3. No Motivation To Modify Because “Arbiter” Is Blind To<br>Command Type .....     | 11          |
| 4. Not “Operative To” .....                                                        | 12          |
| B. No Storage Of Complete Threads .....                                            | 12          |
| C. No “Arbiter” .....                                                              | 13          |
| D. No “Texture Processing Engine” .....                                            | 15          |
| IV. STUTTARD AND WILLIAMS .....                                                    | 16          |
| A. No Storage Of Both “Command Threads” At The Same Time .....                     | 16          |
| B. Stuttard Has No Storage Of “Command Threads” .....                              | 18          |
| C. Stuttard Has No “Arbiter” .....                                                 | 18          |
| D. No Motivation To Add Additional Memory Device To Stuttard .....                 | 19          |
| E. No Motivation To Combine .....                                                  | 19          |
| V. WILLIAMS AND WHITTAKER .....                                                    | 21          |
| A. No Storage Of “Command Threads” .....                                           | 21          |
| B. No “Arbiter” .....                                                              | 22          |
| C. No “Command Processing Engines” .....                                           | 24          |

|     |                                      |    |
|-----|--------------------------------------|----|
| D.  | No “Arithmetic Logic Unit” .....     | 24 |
| E.  | No “Texture Processing Engine” ..... | 24 |
| F.  | No Motivation to Combine .....       | 25 |
| VI. | CONCLUSION.....                      | 26 |

**TABLE OF AUTHORITIES**

**Page(s)**

**Cases**

|                                                                                                            |               |
|------------------------------------------------------------------------------------------------------------|---------------|
| <i>Bettcher Indus., Inc. v. Bunzl USA, Inc.</i> ,<br>661 F.3d 629 (Fed. Cir. 2011) .....                   | 15            |
| <i>Intelligent Bio-Systems, Inc. v. Illumina Cambridge, Ltd.</i> ,<br>821 F.3d 1359 (Fed. Cir. 2016) ..... | 1             |
| <i>Wasica Fin. GmbH v. Cont'l Auto. Systems</i> ,<br>853 F.3d 1272 (Fed. Cir. 2017) .....                  | <i>passim</i> |

**Other Authorities**

|                              |    |
|------------------------------|----|
| 37 C.F.R. § 42.6(a)(3) ..... | 24 |
|------------------------------|----|

**TABLE OF EXHIBITS**

| <b>Exhibit</b> | <b>Description</b>                                                                                                                                                                                            |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2001           | Declaration of Dr. John Hart and CV                                                                                                                                                                           |
| 2002           | Certain Graphics Systems, Components Thereof, and Digital Televisions Containing The Same, Inv. No. 337-TA-1318, Final Initial Determination (July 7, 2023)                                                   |
| 2003           | James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. 1990. Computer Graphics: Principles And Practice (2nd ed.). Addison-Wesley Longman Publishing Co., Inc                                 |
| 2004           | Akenine-Möller, T., Haines, E., & Hoffman, N. (2008). Real-Time Rendering (3rd ed.). A K Peters/CRC Press.                                                                                                    |
| 2005           | Certain Graphics Systems, Components Thereof, and Digital Televisions Containing The Same, Inv. No. 337-TA-1318, Hearing Transcript, Day 4 (March 16, 2023)                                                   |
| 2006           | Edwin Earl Catmull. 1974. “A Subdivision Algorithm For Computer Display of Curved Surfaces.” Ph.D. Dissertation. The University of Utah                                                                       |
| 2007           | Difference Between Buffering<br><a href="https://techdifferences.com/difference-between-buffering-and-caching-in-os.html">https://techdifferences.com/difference-between-buffering-and-caching-in-os.html</a> |
| 2008           | Certain Graphics Systems, Components Thereof, and Digital Televisions Containing The Same, Inv. No. 337-TA-1318, Deposition of Dr. Pedram (January 04, 2023)                                                  |
| 2009           | Declaration of John Hart, Ph.D., Regarding U.S. Patent No. 7,742,053 (Jan. 17, 2024)                                                                                                                          |
| 2010           | Deposition Transcript of Massoud Pedram, Ph.D. (Jan. 8, 2024)                                                                                                                                                 |

|      |                                                                                                     |
|------|-----------------------------------------------------------------------------------------------------|
| 2011 | Frank D. Luna. 2003. Introduction to 3D Game Programming with DirectX 9.0. Wordware Publishing Inc. |
| 2012 | Sumair Ahmed. 2009. “OpenGL – Lighting, Material, Shading and Texture Mapping.” IIT Bombay.         |
| 2013 | Declaration of John Hart, Ph.D., Regarding U.S. Patent No. 7,742,053 (May 22, 2024)                 |

## I. INTRODUCTION

Petitioner's Reply only confirms its Petition fails. Each of the grounds suffers from multiple fatal flaws. Faced with these deficiencies, Petitioner raises over *ten* new arguments asserting obviousness, inherency, and theories relying on previously unidentified portions of prior art, which fail both on the merits and procedurally. The Petition could have, and should have, included these arguments, especially given some of these deficiencies were known from the ITC investigation. As the Federal Circuit provides, “[i]t is of the utmost importance that petitioners in the IPR proceedings adhere to the requirement that the initial petition identify ‘with particularity’ the ‘evidence that supports the grounds for the challenge to each claim.’” *Intelligent Bio-Systems, Inc. v. Illumina Cambridge, Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016). As explained below, even if the Board considers these untimely arguments, they still fail to demonstrate obviousness of the asserted claims.

## II. CLAIM CONSTRUCTION

### A. Storage of “command threads”

To meet the claim requirements of the '053 patent to store a command thread, a complete thread must be stored at a given moment in time. POR, 7-8. Petitioner made this very argument to the ITC, but attempts to run from it now. Ex. 2002, 32-33 (ALJ explaining Petitioner argued “a complete thread must be stored at a given moment in time” and disputed that “portions of a complete thread are sufficient so

long as the entire thread is stored over a period of time”). The ALJ agreed with Petitioner. *Id.* Nothing in the Reply attempts to respond.

Instead, Petitioner argues for the first time it would be obvious to store complete command threads. *Id.* But as explained in Section III.B, this obviousness argument was waived because Petitioner failed to present it in its Petition. *Wasica Fin. GmbH v. Cont’l Auto. Systems*, 853 F.3d 1272, 1285-87 (Fed. Cir. 2017) (affirming Board’s ruling an obviousness challenge was “insufficiently precise and underdeveloped” because petitioner “did not make out its obviousness case in its petition”).

Petitioner also claims Dr. Hart lacks credibility. But Dr. Hart’s refusal to agree with Petitioner’s counsel’s questioning was because it was unclear. Dr. Hart repeatedly asked for clarification, which counsel refused to give, and as a result, Dr. Hart testified: “I don’t follow that question. I don’t -- I can’t agree with -- I can’t answer yes to that. I’d say no.” Ex. 1016, 32:4-33:20, 34:7-21. This does not prove Petitioner’s point, nor address the facts above. Further, Dr. Hart was consistent in his testimony that complete command threads must be stored. *E.g., id.*, 31:4-16, 31:20-32:3, 35:8-15.

**B. “at least one memory device . . .”**

The “memory device” must comprise “a first portion operative to store . . . pixel command threads and a second portion operative to store . . . vertex command

threads.” Ex. 1001, claims 1, 5. Petitioner’s insistence that both command thread types do not need to be stored simultaneously renders the “operative to store” language meaningless. POR, 32-33. Even if “operative to” means “capable of” as Petitioner alleges, the arbiter cannot select from both command thread types if they are not both simultaneously stored. The specification only provides examples where the memory device stores both command thread types simultaneously. *Id.*, 9 (listing examples). Petitioner fails to address this.

**C. “select a command thread . . .”**

Petitioner argues that Patent Owner reads out “either of” by requiring pixel and vertex command threads to be simultaneously stored. Nonsense. The arbiter must be operable to select “from *either of* the plurality of pixel command threads *and* the pluralities of vertex command threads.” Ex. 1001, claims 1, 5 (emphasis added). Both types must plainly be present to be selected from. POR, 9-11.

**D. “command processing engine”**

Petitioner claims that certain command processing engines in the plurality only need to process vertex command threads, while others only need to process pixel command threads. That contravenes the claims and divorces the specification from the language of the claims. POR, 11-13. Dr. Pedram agreed, at least for claim 5, each command processing engine must be operable to process whichever

command thread is provided to it by the arbiter, be it pixel or vertex. Ex. 2008, 203:1-18.

### **III. STUTTARD**

#### **A. No Storage Of “Command Threads” At The Same Time, In Separate Portions**

Petitioner argues that the claims do not require storing pixel command threads and vertex command threads simultaneously. Reply, 4. If so, the arbiter would not be able to select from the pluralities of both command thread types. POR, 7-11. The ALJ agreed. Ex. 2002, 84 (“[I]t is a correct claim construction that [the “at least one memory device” limitation] means the memory device(s) are operable to store both pluralities at the same time.”).

Petitioner misrepresents Dr. Hart’s testimony in asserting he formed no opinions on whether Stuttard’s memory was capable of storing both pluralities at the same time. Reply, 4. Rather, Dr. Hart was questioned regarding capability versus actual configuration and testified that he applied the claim language “operative to” in his opinions, and further opined that “operative to” required actual configuration. Ex. 1016, 58:12-59:20.

Nor did Dr. Hart admit he did not consider what the typical size of a cache memory is. Rather, he testified: “[i]t’s a cache, meaning it’s designed to hold a small amount of information that can be accessed quickly. It’s not designed to hold an

entire thread; it's designed to hold an instruction so that that instruction can be accessed quickly.” *Id.*, 57:7-58:11.

Further, a POSITA would not be motivated to modify Stuttard to store pixel and vertex command threads at the same time because of phased nature of Stuttard's graphics processing and SIMD design. POR, 21-31. This does not change even with Stuttard's segmented FIFO. While Stuttard discloses its cache memory could be replaced with FIFOs, one per thread, this says nothing about the thread types being stored. Ex. 1005, 11:5-7. Given Stuttard's phased nature and SIMD design, the individual buffers of the FIFO will store pixel command threads or vertex command threads, but not the two types together. Further, the use of FIFOs means the arbiter does not select from “a plurality” of both command thread types, as required by the claims, because a system can only receive what comes out at the end of the FIFO—it does not select anything. Ex. 1016, 80:19-82:14; POR, 38-39.

Petitioner also identifies the APA's FIFO buffers as disclosing the storage of both thread types simultaneously. Reply, 5. As explained in the previous paragraph, the same issues with Stuttard's FIFOs apply equally here. Ex. 1016, 80:19-82:14; POR, 38-39.

### **1. Phased Operation**

Petitioner claims Patent Owner fails to consider Stuttard as a whole. Reply, 6. But it is Petitioner who takes a microscopic view of Stuttard. Not once did

Petitioner identify any disclosure in Stuttard which provides its processing blocks can process different threads or instructions at the same time. In fact, Petitioner repeatedly identifies the same disclosure to argue Stuttard executes different tasks at the same time. Reply, 6-7 (both citing Ex. 1005, 11:16-25). But Petitioner fails to address that the disclosure relates only to what the cache unit of the thread manager stores, which is not part of the processing elements or processing blocks. Thus, even though Stuttard says “at any particular moment in time each thread may be assigned to any one of a number of tasks,” this relates only to their storage, and not their execution. Ex. 1005, 11:16-25; Ex. 2013 (Hart Sur-Reply Decl.), ¶2.

There are multiple reasons why Stuttard would store instructions from different threads despite only executing them one at a time, such as preventing downtime. As Stuttard describes, the system will want to switch from one thread to another thread if the current active thread stalls. Ex. 1006, 23:35-25:31. This switch prevents downtime, and storing multiple threads at the same time, and also processing them via the thread processors, ensures the next thread is ready to start executing when the active thread stalls. Ex. 2013, ¶3. Further, Stuttard time-interleaves instructions from different threads for execution. *Id.* Having more thread processors enables the interleaving of instructions for multiple threads, which also helps avoid stalls.

Petitioner handwaves Patent Owner's arguments regarding binning units by claiming Patent Owner ignored the fact that there is one per processing block. Reply, 7. But Stuttard's entire system is set up so the array of processing elements must wait for the locations of the triangles to be determined (which is done by the binning units) before moving onto pixel processing. POR, 21-26. The fact that each processing block has a binning unit does nothing to change this fundamental operation. Further, Petitioner failed to address Stuttard's express disclosure that both the geometry phase and rasterization phase are "block parallel," and that each binning unit transfers data between the different processing blocks. Ex. 1005, 41:14-17 (describing the "*need*[]" for a mechanism transferring data between any block to any of the bins and between any bin and any block"). If Petitioner was correct that each processing block and its binning unit operated independently, each processing block would need to process geometry separately, negating the benefits of parallelism. Ex. 2013, ¶4. If geometry processing was divided across different independent blocks, then a triangle in one block may overlap a portion of the screen processed by another block, which had already made it to the pixel processing phase. *Id.* This would hinder the process of determining which triangles show up on the screen (done via binning across all the processing blocks) and may result in unnecessary processing of triangles which do not show up, a waste of resources. *Id.*

Petitioner argues for the first time it would have been obvious for Stuttard to *process* (as opposed to store) both thread types at the same time. Reply, 6-7. This argument is waived because Petitioner did not argue it in its Petition. Pet., 26-27; *Wasica*, 853 F.3d at 1285-87. Even if not waived, Stuttard does not render this obvious because Stuttard discloses only phased operation. There is no motivation to modify Stuttard to process both thread types simultaneously.

Lastly, Petitioner provides some examples it believes contradicts Stuttard's phased operation. Reply, 7. First, Petitioner hypothesizes some processing blocks will be idle due to the lack of any pixel command threads. But if the processing is spread evenly throughout the entire system and operating on different groups of data items, as Stuttard does, then the blocks will not be idle. Ex. 2013, ¶5. Stuttard provides it resolves this very issue by organizing the processing elements array into processing blocks, "the processing blocks being operative to process respective groups of data items." Ex. 1005, 1:24-2:23. Second, Petitioner posits a video frame with updates in two different regions, one with more intensive vertex processing and one with less intensive vertex processing also will render some processing blocks idle. Ex. 2013, ¶5. But this is not so, because the entirety of Stuttard's system would finish vertex processing of both areas before switching to pixel processing of both areas given the binning units maintain both geometry phase and rasterization phase

are block parallel. This means all the processing blocks are either doing geometry or rasterization, but not both.

## **2. SIMD**

Petitioner claims finding Stuttard has a single SIMD is “fatal flaw.” Reply, 7-8. But not once has Petitioner identified any disclosure in Stuttard providing it has multiple SIMDs. Nor has Dr. Pedram. To the contrary, Stuttard itself expressly discloses it is a single SIMD. POR, 26-28. And Dr. Pedram recognizes Stuttard comprises “a SIMD” array of processing elements, where the array of PEs is divided into processing blocks. Ex. 1004, ¶51 (citing Ex. 1005, 2:17-23). Thus, all of the processing blocks are part of the same single SIMD. Stuttard never states each processing block is a separate SIMD.

Here again, Petitioner identifies 11:15-35 as supporting the notion that Stuttard is capable of executing more than one instruction (and hence one thread) at a time. Again, this disclosure only relates to the storage of threads in Stuttard’s cache memory. Petitioner also identifies Stuttard’s desire to keep the processing blocks as active as possible. But this desire is satisfied by Stuttard’s ability to switch between threads when one thread becomes inactive, and by Stuttard’s use of interleaving of instructions as explained above in Section III.A.1. *See also* Ex. 2013, ¶3. Thus, if a thread is stalled, then that thread is swapped. It does not establish the processing blocks can be executing *different* instructions or threads.

Petitioner asserts Patent Owner's arguments that the processing blocks operate only on different data, not different instructions, are wrong. But Petitioner fails to account for Stuttard's explicit disclosure that, after binning, "a given processing block operates on data associated with a single region of the display." Ex. 1005, 22:6-10; POR, 21-26. Petitioner's argument that this discussion focuses only on one processing block is incorrect. This disclosure is from the section discussing Stuttard's graphics processing procedure as a whole and applies to the entire array of PEs in Stuttard's system, and thus all of the processing blocks within that array. *Id.*, 20:21-33 ("Figure 9 illustrates simplified steps in a graphics data processing method. . . . The data is loaded into *the PEs of the graphics system* so that each PE contains data for one vertex."). Further, Stuttard explicitly states that its single SIMD array of processing elements "are operably divided into processing blocks, the processing blocks being operable to process respective data items." *Id.*, 2:17-23. This disclosure does not suffer Petitioner's alleged deficiencies regarding the binning units.

Petitioner's argument that Stuttard discloses additional PEs for redundancy does not undermine Patent Owner's argument that the processing blocks provide additional redundancy. POR, 30. Stuttard provides for each panel of 32 PEs, one additional PE is provided. Ex. 1005, 9:30-10:3. This doesn't negate the fact that grouping blocks of PEs into processing blocks also provides additional redundancy

in case the failure is greater than a few PEs. Further, having multiple redundancy points is beneficial to a system. Ex. 2013, ¶6.

Finally, putting aside the fact that Petitioner has waived any reliance on claims 103 and 104, Petitioner's assessment of claims 103 and 104 is wrong and divorced from the language of the claims. The claims generally pertain how *data items* are accessed by *the single SIMD array of processing elements*. They do not disclose how *instructions* are executed by the SIMD array. POR, 30-31. They also do not disclose different instructions relating to different data items or different instructions being processed at the same time.

### **3. No Motivation To Modify Because “Arbiter” Is Blind To Command Type**

Petitioner argues Stuttard does not need to care about command type to execute different thread types concurrently. Reply, 10. But this was not Patent Owner's argument. Patent Owner's argument was in rebuttal to Petitioner's motivation to modify Stuttard's cache unit to have different portions for storing the different thread types. POR, 31-32.

Further, Petitioner provides no support for its assertions from either Stuttard or from its expert. Nor does it provide any support for the proposition that the thread processors “launch” threads. It is actually the thread scheduler which schedules threads for execution. Ex. 1005, 12:15-18.

#### 4. Not “Operative To”

Petitioner has not shown Stuttard is capable of or configured. POR, 32-33. Petitioner attempts to distinguish *Aspex* by claiming the patent there was using the narrower meaning, but fails to explain why. Reply, 10.

Even if “operative to” and “operable to” only means “capable of,” Petitioner has not met that showing because Stuttard’s cache memory *never* stores pixel and vertex command threads at the same time, nor does the alleged arbiter *ever* select between at least two pixel and at least two vertex commands. POR, 21-26, 36-39.

#### B. No Storage Of Complete Threads

Petitioner claims Stuttard never says it is storing “partial” command threads. However, Stuttard never states that it stores complete threads either. A close look at Stuttard’s disclosures reveals it is instructions which are fetched by the thread manager from the host system. POR, 33-36; Ex. 1005, 8:15-21. And it is these instructions which are ultimately provided by the thread manager to the array controller and channel controller, and finally to the PE array. *Id.*, 8:17-24. As explained above, Dr. Hart further explained from a technical perspective why it is clear that a cache only stores partial command threads, which Dr. Pedram agreed with in the ITC. Ex. 1016, 57:7-58:11; Ex. 2008, 183:6-10.

Petitioner’s argument it is obvious Stuttard can store complete threads should be disregarded because it was never made in the Petition. Reply, 11; *Wasica*, 853

F.3d at 1285-87. Even if not waived, Petitioner provides no explanation why it is obvious, nor why a POSITA would be motivated to modify Stuttard in this way given that Stuttard operates on an instruction level so there is no need to store the entirety of the thread. POR, 33-36.

With respect to Petitioner's assertion there was no citation to Dr. Pedram's contradictory testimony, any omission was accidental. Dr. Pedram testified the arbiter must recognize the type of command thread it is selecting at Ex. 2010, 217:17-218:9. Dr. Pedram testified he didn't know whether Stuttard's arbiter knew the command type it selected is at Ex. 2010, 318:11-319:17. POR, 31-32. While Petitioner contests Dr. Pedram opined Stuttard was limited to prefetch and execute operations, the POR expressly quotes Dr. Pedram's testimony. POR, 34-35 (citing Ex. 2008, 183:1-4). Dr. Pedram's testimony that this is "typically what happens in *these kind of systems*" is in reference to Stuttard. Ex. 2008, 181:15-21 (questioning attorney asking question regarding Stuttard), 183:1-4 (providing answer in these kinds of systems). It is Petitioner who provides no evidence to support its assertion.

### **C. No "Arbiter"**

Petitioner argues the APA renders "storing both pluralities at the same time and selecting between both pixel and vertex command threads" obvious. Reply, 12. But Petitioner never relied on the APA in its Petition for this limitation. Pet., 25-26,

32-35 (only citing to Stuttard). Such arguments are waived. *Wasica*, 853 F.3d at 1285-87.

Even if not waived, the APA provides only an ALU resource and texture fetch resource, it does not disclose pixel command threads and vertex command threads. POR, 38-39. Further, the APA uses FIFOs. *Id.* Even assuming *arguendo* that the arbiter is not required to select between both pixel and vertex command threads, as Petitioner argues, the arbiter cannot select from *a plurality* of pixel command threads or *a plurality* of vertex command threads because APA's FIFO buffers are first-in-first-out, meaning at best only one command thread would be available and thus there is no selecting from among multiple command threads. *Id.*; Ex. 1016, 81:3-18. Thus, the "pluralities" requirement of the claim is not satisfied.

Further, Petitioner misrepresents Dr. Hart's testimony. Dr. Hart did not admit the APA discloses it selecting between plurality of pixel and vertex command threads. Reply, 13. Rather, Dr. Hart testified that arbiter 102 can receive from different *FIFO buffers*, but the posed question asked nothing about pixel and vertex command threads, much less selecting from a plurality of them, which a system cannot do from a FIFO. Ex. 1016, 16:16-19. Dr. Hart further confirmed the FIFOs do not satisfy the claims because they: (i) do not enable selection of a plurality of anything since they are first-in-first-out, meaning only one item is available for

selection, not a plurality, and; (ii) the FIFOs do not store pixel and vertex command threads. *Id.*, 80:19-82:14.

Lastly, Petitioner argues for the first time that selecting between both command thread types is obvious. This argument is waived as it was not disclosed in the petition. POR, 25-26. *Wasica*, 853 F.3d at 1285-87. Even if not waived, Petitioner has provided no substantive argument as to why it is obvious.

**D. No “Texture Processing Engine”**

Petitioner’s argument that this limitation is obvious is waived because it never presented it in the Petition. Reply, 14; *Wasica*, 853 F.3d at 1285-87. Petitioner only argued inherency by claiming a POSITA would recognize that lighting and shader “necessarily” requires a texture processing engine. Pet., 27-28, 39.

Petitioner’s inherency argument fails because inherency cannot be established by “probabilities or possibilities.” *Bettcher Indus., Inc. v. Bunzl USA, Inc.*, 661 F.3d 629, 639 (Fed. Cir. 2011). Processing lighting and shading does not *necessarily* involve a texture processing engine (a required element to show inherency), and a POSITA would have considered texture processing a separate step from lighting and shading in graphics processing. POR, 39-40.

#### **IV. STUTTARD AND WILLIAMS**

Petitioner relies only on Williams in this combination to provide the storage of different command threads by type in separate portions of memory. Reply, 14, 17, 18-19.

##### **A. No Storage Of Both “Command Threads” At The Same Time**

Petitioner’s reason for modifying Williams creates a solution without a problem. Petitioner argues storing different types of command threads in different portions of memory would streamline the transference of graphics commands to different buses. Reply, 15. But this “streamlining” is unnecessary because Williams’ graphics commands are already tagged by type. POR, 47. While Petitioner urges its modification is “obvious and not cost prohibitive,” a POSITA would not be motivated to add this redundancy into the system when the pre-existing tagging system provides far more flexibility. POR, 45-50. A POSITA would understand that any additional information (especially unnecessary and duplicative) would decrease system efficiencies, especially given the extra information needed to be stored across all the different commands (of which there are likely a large multitude per frame). *Id.*

Petitioner argues Williams does not disclose that graphics commands stored in the buffer are retrieved in the same order they went in. Williams provides “[i]nput buffer 202 stores the graphics commands such that their original order (i.e., the order

in which the graphics command were transferred from the system bridge 106 to input buffer 202) is known.” Ex. 1006, 11:22-24. Williams also explicitly discloses preservation of the graphics commands’ original order. POR, 67-68. The graphics bus de-scheduler (GBD), which functions similarly to the GBS and located on the other end of the busses, accepts the multiple graphics commands that are transferred across the plurality of busses. *Id.*, 11:5-8, 14:8-11. If after transferring, the graphics commands “do not necessarily have the specific order” from the input buffer, GBD will put them back in the same order. *Id.*, 14:8-21. “GBD 210 then transfers the accepted multiple graphics commands into output buffer 212, according to the tags, such that the multiple graphics commands regained the specific order (that is, their original order) within output buffer 212.” *Id.* In fact, if one of the graphics commands is missing, GBD can put a placeholder in output buffer 212 so that when it shows up, it can be placed in the correct position in the sequence. *Id.* Petitioner failed to address this specific disclosure when asserting the graphics commands are not received in the same order. Reply, 16. For this reason, Petitioner’s purported benefit of having consecutive addresses for graphics commands by type would be a hindrance to Williams’ system.

Petitioner states for the first time it is relying only on Stuttard’s arbiter for this ground. Reply, 17. Therefore, the same reasons Stuttard doesn’t disclose the arbiter for ground 1 also apply here. POR, 36-39. Further, with respect to Petitioner’s

argument that Stuttard benefits from partitioning memory into different portions for each command type, as explained in the POR and above, Williams does not suggest this or make it obvious. Stuttard does not benefit from partitioning memory into different portions for each command type because it is a SIMD and because it executes one instruction at a time, and because its arbiter does not care about command type as Dr. Pedram testified. Ex. 2010, 318:11-319:17.

**B. Stuttard Has No Storage Of “Command Threads”**

Petitioner states for the first time it relies solely on Stuttard for storing command threads for ground 2. Reply, 17. Thus, all of Stuttard’s deficiencies apply to this ground. Further, Petitioner admits Williams stores OpenGL commands, meaning Williams does not disclose command threads. *See* Section V.A, *infra*.

Further, Petitioner relies on the Board’s theory that the accumulation of commands in display list 704 discloses complete command threads. Besides being incorrect, Petitioner cannot rely on display list 704 because it never identified it as a “memory device” in the Petition and it is thus waived. Pet., 40-43, 46-48; POR, 57-58; *Wasica*, 853 F.3d at 1285-87. Further, Stuttard does not store complete command threads. *See* Section III.B, *supra*; POR, 33-36.

**C. Stuttard Has No “Arbiter”**

Petitioner states Patent Owner did not identify any issues with Stuttard’s arbiter. That is untrue. POR, 36-39.

**D. No Motivation To Add Additional Memory Device To Stuttard**

Petitioner argues using multiple memory devices is a simple design choice. Assuming *arguendo* that this is true, it does not in and of itself present a motivation to modify Stuttard in that manner. As Patent Owner explained, there is no motivation. POR, 51-52.

**E. No Motivation To Combine**

Petitioner claims combining Stuttard and Williams provides a “host of benefits.” Reply, 18-19. However, this argument is unsupported by Petitioner’s theories because as Petitioner states *for the first time*, all it is taking from Williams is the idea of partitioned memory. *See id.* Thus, Petitioner’s arguments relating to the benefits of increased speed by virtue of using Williams in lieu of the AGP bus in Stuttard, such as greater throughput, are moot. Pet., 45-46. As are any purported benefits from the use of Williams’ system bridge in Stuttard. *Id.*

If Petitioner had clearly stated its theory from the outset, Patent Owner would have been able to properly respond to it. Case in point, the Board thought Williams’ bus lines were being added to Stuttard, which Petitioner now assures it is not. Reply, 19 (stating “Petitioner does not require that Williams’ dedicated bus architecture be incorporated into Stuttard”); ID, 42-45. Similarly, the Board thought Williams’ GBS was being added to Stuttard as well, which Petitioner also denies. ID, 44-45; Reply, 17 (asserting “no issue[s]” with “Stuttard’s arbiter” and it “needs no

substantive modification”). Petitioner’s lack of narrative and details fail to provide notice regarding its theories.

Petitioner failed to address Patent Owner’s argument that Stuttard and Williams are incompatible. POR, 51-52; Reply, 18-20. Petitioner claims that increasing bandwidth (as Williams is directed to) and increasing graphics processing speed (as Stuttard is directed to) are related concepts. Reply, 18-19. These are unrelated because the first focuses on shuttling data from host system to another system (e.g., a graphics system), whereas the latter focuses on increasing the processing capacity. POR, 60-61. This does not provide a motivation to modify Stuttard because Stuttard does not need to increase bandwidth. *Id.*

Petitioner asserts it is a red herring to argue bandwidth is a limiting factor in Stuttard. Reply, 19. But Patent Owner did not argue such. Patent Owner stated “Stuttard’s system is not limited by the bandwidth of the AGP buses.” POR, 61. Even if Patent Owner did argue this, it is supported by Williams’ express disclosure: “peripheral bus 114 is often a bottleneck in computer system.” Ex. 1006, 3:27-4:5.

Lastly, Petitioner’s argument at Reply, 19 that adding more processing blocks in Stuttard would result in a desire to increase data transfer rates and provide a motivation to combine the two references is waived as it was not presented in the Petition. *Wasica*, 853 F.3d at 1285-87.

## **V. WILLIAMS AND WHITTAKER**

Petitioner relies only on Whittaker in this combination to provide the “relative priorities” limitation. Pet., 53.

### **A. No Storage Of “Command Threads”**

Williams does not render storing command threads obvious. Petitioner did not allege this in its Petition; thus it is waived. *Wasica*, 853 F.3d at 1285-87. The only obviousness arguments made with respect to the Williams/Whittaker combination relate to the arbiter and priority. Pet., 51, 54-55.

Petitioner admits Williams’ graphics commands are in fact OpenGL commands, and not the required command threads. Reply, 17 (stating “Williams’ OpenGL commands are high level commands.”). Even if Williams’ higher-level commands “give rise to” vertex or pixel processing, this argument is waived, as it was never alleged in the Petition. *Wasica*, 853 F.3d at 1285-87. Further, Petitioner fails to explain how or why these high-level commands give rise to vertex or pixel processing, especially in light of Patent Owner’s arguments they do not. POR, 53-58, 65-66.

Petitioner also does not address Patent Owner’s arguments as to why Williams’ output buffer, which Petitioner alleges stores vertex commands, cannot be one of the claimed memory devices. POR, 48-49 (explaining output buffer

receives transferred commands, after the GBS's purported selection applies to Ground 3), 65-66.

**B. No “Arbiter”**

By arguing the GBS “selects,” Petitioner impermissibly broadens the plain and ordinary meaning of “select” to include “sort” and “assign,” contrary to its own expert's testimony. Dr. Pedram stated: “[a]n arbiter selecting a command thread means that it looks at command threads that are available and chooses one, selects one. That's the meaning a person skilled in the art would understand the function of the arbiter to be in this context, also consistent with the '053 specs.” Ex. 2010, 196:24-197:4.

Williams' GBS does not disclose the claimed arbiter because it does not choose one or more items from a group of items, which is the plain and ordinary meaning of “select.” Ex. 2013, ¶7. This is further supported by Dr. Pedram's additional testimony that an arbiter which “ha[s] nothing to do with what inputs arrive at its input terminals” does not have a “selection of choice.” Ex. 2010, 205:17-20. The “sorting” and “assigning” identified by Petitioner is actually the GBS “receiv[ing]” the next graphics command which exits the input buffer and putting it on one of the bus lines, not selection. Ex. 1006, 18:26-28. Thus, the GBS only acts upon any graphics command once it has already left the input buffer.

Williams' GBS is also not an arbiter is because the graphics commands leave the input buffer in the same order they enter. POR, 67-68; *see* Section IV.A, *supra*. Thus, Williams' GBS has no "selection of choice" (in Dr. Pedram's parlance) which command exits the input buffer.

Petitioner argues for the first time that the arbiter is not required to "select from" the memory device; this argument is waived. Reply, 21; *Wasica*, 853 F.3d at 1285-87. Even still, the language of the claim requires it. Ex. 2013, ¶8. The arbiter must "select a command thread *from* either of *the* plurality of pixel command threads and *the* plurality of vertex command threads." Ex. 1001, Claims 1, 5. The reference of "the" refers back to the command threads stored by the memory device. Meanwhile, the different assignment schemes identified by Petitioner do not disclose selection. Reply, 21-22. Rather, the assignment schemes merely provide what the GBS does after it has already received a graphics command (*i.e.*, if there was selection, what would happen after the selection).

Lastly, Petitioner asserts that Whittaker's MCC discloses "selecting" and the arbiter requirement. But Petitioner relied on Whittaker only for the "relative priorities" requirement of the claim, thus this new argument is waived. Pet., 53 ("Specifically, in combination, the priority information of Whittaker is provided to Williams' GBS, which, when assigning command threads to buses according to

command type, makes the assignment based on the priority information of the respective threads.”); *Wasica*, 853 F.3d at 1285-87.

**C. No “Command Processing Engines”**

Petitioner has not shown the alleged command processing engines are operable to receive and process both command thread types. POR, 11-13, 68-69; Ex. 2008, 203:1-18.

**D. No “Arithmetic Logic Unit”**

Petitioner provided *no* explanation in its Petition as to why this claim is disclosed or rendered obvious. Pet., 60; POR, 69-70. Petitioner asserts it cited to Dr. Pedram’s testimony. This violates 37 C.F.R. § 42.6(a)(3)’s prohibition on incorporating arguments by reference from one document (Dr. Pedram’s declaration) into another document (the Petition). Even accounting for Dr. Pedram’s opinion, he gives only conclusory opinion to assert a POSITA would recognize Williams includes an arithmetic logic unit. Ex. 1004, ¶148 (asserting geometry operations “necessarily” disclose arithmetic logic units without explaining why”).

**E. No “Texture Processing Engine”**

Petitioner provided no explanation in Petition, and any allegation those arguments were made in Dr. Pedram’s declaration also violates 37 C.F.R. § 42.6(a)(3). Pet., 61; POR, 70. Petitioner now imports the sentences from Dr.

Pedram's declaration into its Reply, but since those arguments were not made in the Petition, it is waived. Even if not waived, Dr. Pedram's opinion is conclusory.

**F. No Motivation to Combine**

As with Ground 2, Petitioner characterizes Williams and Whittaker as merely having different approaches. But the different approaches are directed *different problems* (increasing bandwidth as in Williams versus increasing processing speed as in Whittaker) in *different media sectors* (3D graphics processing as in Williams versus video processing as in Whittaker). POR, 63-65, 70-71. While Petitioner claims Whittaker is directed to 3D graphics processing, its support is lacking. Ex. 1004, ¶138 (citing Ex. 1007, 1:9-20, 6:50-56). The first citation is in the background and is unrelated to Whittaker's system. The second citation is actually focused on video processing, identifying the inclusion of several components that are only used in video processing. POR, 64-65, 70-71. Thus, Whittaker is focused video processing, not 3D graphics processing.

Lastly, Petitioner's assertion "a POSITA reading the references would know . . . Whittaker's teachings can be applied to Williams' system . . ." is conclusory without providing any explanation. Reply, 24. And Dr. Pedram does not provide any additional explanation either, merely parroting the Reply. Ex. 1015, ¶47.

## VI. CONCLUSION

For the foregoing reasons, the Board should determine Petitioner has failed to prove any claims of the '053 patent are unpatentable.

Dated: May 22, 2024

Respectfully submitted,

/s/ William A. Meunier

William A. Meunier (Reg. No. 41,193)

Michael T. Renaud (Reg. No. 44,299)

Adam S. Rizk (Reg. No. 66,867)

MINTZ, LEVIN, COHN, FERRIS,  
GLOVSKY AND POPEO, P.C.

One Financial Center

Boston, MA 02111

Telephone: 617-542-6000

Facsimile: 617-542-2241

E-mails: WAMeunier@mintz.com;

ARizk@mintz.com

MTRenaud@mintz.com

Reza Dokhanchy (Reg. No. 62,795)

MINTZ, LEVIN, COHN, FERRIS,  
GLOVSKY AND POPEO, P.C.

3580 Carmel Mountain Road, Suite 300

San Diego, CA 92130

Telephone: 858-314-1596

[RDokhanchy@mintz.com](mailto:RDokhanchy@mintz.com)

**CERTIFICATE OF WORD COUNT**

Pursuant to 37 C.F.R. §42.24(d), Patent Owner hereby certifies, in reliance on the word count of the word-processing system (Microsoft Office Word 2010) used to prepare this preliminary response, that the number of words in this paper is 5,590. This word count excludes the tables of contents, tables of authorities, certificate of word count, certificate of service, and table of exhibits.

Dated: May 22, 2024

/s/ William A. Meunier

William A. Meunier (Reg. No. 41,193)

**CERTIFICATE OF SERVICE**

I certify that a copy of Patent Owner's Sur-Reply to Petitioner's Reply and accompanying Exhibit 2013 are being served by electronic mail on the following counsel of record:

**Lead Counsel**

Jeffrey Johnson  
Reg. No. 53,078  
BAKER BOTTS, LLP  
910 Louisiana Street  
Houston, TX 77002-4995  
Telephone: (713) 229-1234  
Facsimile: (713) 229-1522  
Email:  
Jeffrey.Johnson@bakerbotts.com;  
DLBBRealTEKIPRs@bakerbotts.com

**Backup Counsel**

Christopher J. Higgins  
Reg. No. 66,422  
ORRICK, HERRINGTON &  
SUTCLIFFE, LLP  
2100 Pennsylvania Avenue NW  
Washington, DC 20037  
Telephone: (202) 339-8400  
Facsimile: (202) 339-8500  
Email: 0CHPTABDocket@orrick.com;

Steve Baik  
Reg. No. 42,281  
WHITE HAT LEGAL  
P. O. Box 382  
San Jose, CA 95002  
Telephone: (650) 618-5282  
Email: sbaik@whitehat.legal

Dated: May 22, 2024

/s/ William Meunier

William Meunier (Reg. No. 41,193)